Problem E: Counting Primes

Achilles and the Tortoise are walking back to their homes after spending the afternoon at the movies, where they watched "Forrest Gump".

Tortoise: That Forrest really could run. Right, Achilles?

Achilles: Certainly. Oh, that reminds me, I wanted to tell you something, Mr. T. Sorry if I give you a hard time with all that “fleetest of all mortals” and “ploddingest of all plodders” stuff. Truth is, I admire you very much. Even if you’re not the fastest on foot, you’re certainly the fastest I know with mental calculations.

Tortoise: Oh, that’s okay, Achilles, it has never bothered me. Maybe I can’t run as fast as you because I was born with these short legs and there’s not much I can do about it, but I’m in peace with that. But you know what’s nice? You and I have something that knows no limitations, and that is our minds. And I think that with a little exercise, you certainly can improve your mental abilities.

Achilles: Hmm. You make a good point, Mr. T. And I do try to exercise my mind. I solve *sudokus* every morning...

Tortoise: Heheheh. I won’t comment on that. But may I suggest another little game that can help your memory and concentration, and becomes useful if you work in number theory problems?

Achilles: Sounds great. What is that?

Tortoise: It’s simply called “counting primes”. I give you a list of N positive integer numbers, and you memorise it.

Achilles: Okay, I can do that.

Tortoise: That’s not all. Then I ask you to perform a series of operations, which can be of two types. I could ask you: how many prime numbers are there between the ith and the jth number in the list? And you would have to answer me correctly. Or I could ask you: replace all numbers in the range i to j with this new number x.

Achilles: That’s... a little harder. But I want to try it.

Tortoise: Alright. We can start easy, with short lists, but we’ll use larger lists later on, okay?

Achilles: Go ahead, I’m ready.

Write a program that can play the *counting primes* game with large lists.

Input

Input starts with a positive integer **T**, that denotes the number of test cases. The first line contains two integers: **N** and **Q**, which represent the size of the array and the number of operations to perform.

The next line contains N integers $a_1, a_2, a_3, \ldots, a_N$, separated by single spaces, which form the original list of numbers.

Q lines follow. Each of these lines comes in one of the following two alternatives:

- **0 *i j x*** – means that all numbers from a_i to a_j (inclusively) are to be replaced with the number x.
- **1 *i j*** – means that you should report the total amount of prime numbers between a_i and a_j, inclusively.

$T \leq 10$; $1 \leq N \leq 10^4$; $1 \leq Q \leq 2 \times 10^4$; $2 \leq a_i \leq 10^6$; $1 \leq i \leq j \leq N$; $2 \leq x \leq 10^6$

Output

For each test case, print the case number in a single line. Then, for each operation of type **1**, print its result in a line of its own.

Sample Input	Output for Sample Input
1 5 3 78 2 13 12 3 1 1 2 0 4 4 5 1 1 5	Case 1: 1 4

Notes

- The test data is large. Make sure to use fast I/O methods.
- A prime number is a natural number greater than 1 that has no positive divisors other than 1 and itself. The first prime numbers are $2, 3, 5, 7, 11, \ldots$